

# Les Dix Commandements de Yoda

*Louis-Marc Mercier*

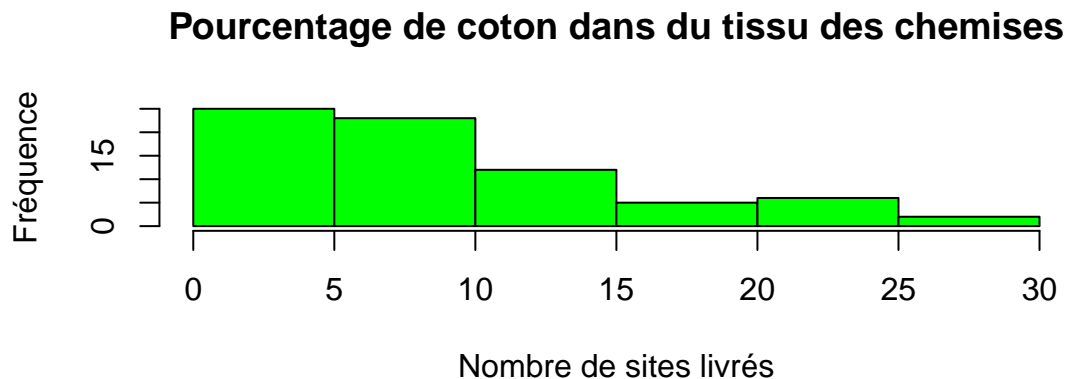
*25 février 2017*

**Premier commandement:** Tu utiliseras la fonction `read.csv()` pour lire un jeu de données en fichier csv.

```
Sites_web <- read.csv("C:/Users/Louis-Marc/Desktop/Travaux scolaires/Universite/Maitrise/Session 5/Charp")
```

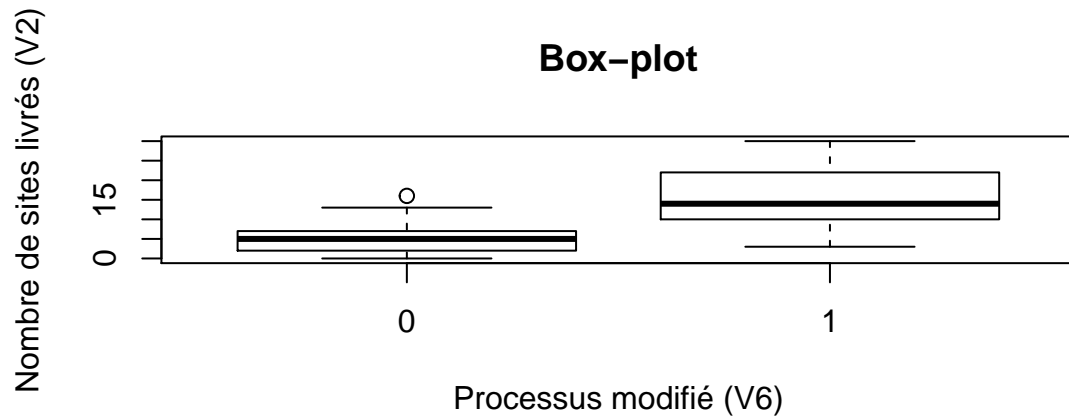
**Deuxième commandement:** Te ne feras pas d'histogramme avec une autre fonction que `hist()`.

```
hist(Sites_web$V2,main="Pourcentage de coton dans du tissu des chemises",  
xlab="Nombre de sites livrés",ylab="Fréquence",col="green")
```



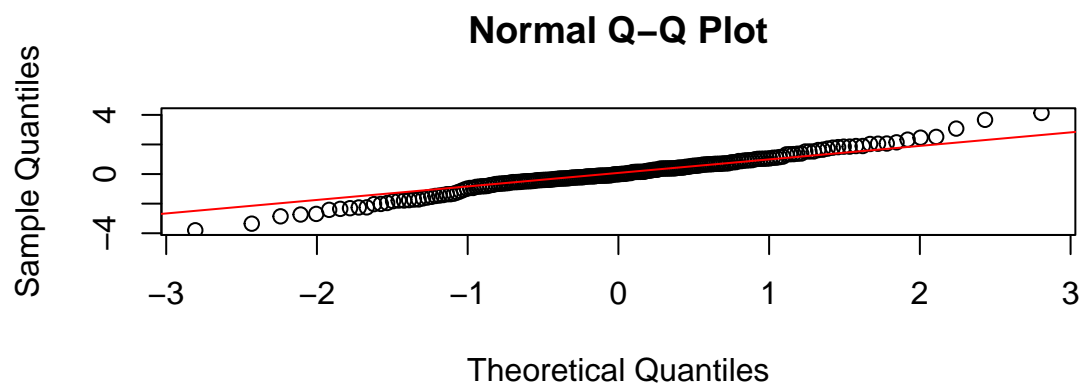
**Troisième commandement:** La fonction `boxplot()`, tu l'utiliseras pour tracer des diagrammes en boîte à moustaches.

```
boxplot(Sites_web$V2~Sites_web$V6,data=Sites_web, main="Box-plot",  
xlab="Processus modifié (V6)", ylab="Nombre de sites livrés (V2)")
```



Quatrième commandement: Pense à utiliser les fonctions `qqnorm()` et `qqline()` pour analyser les résidus.

```
y <- rt(200, df = 5)
qqnorm(y); qqline(y, col = 2)
```



Cinquième commandement: Sépare ton jeu de données avec `subset()` pour jouir d'une manipulation plus facile des données.

```
Sites_web_2001 = subset(Sites_web, V7==2001)
Sites_web_2002 = subset(Sites_web, V7==2002)
```

Sixième commandement: Tu utiliseras les fonctions `mean()`, `var()` et `mode()` dans le but de calculer la moyenne, la variance et le mode.

```
x = seq(1:10)
moyenne = mean(x)
variance = var(x)
le_mode = mode(x)
```

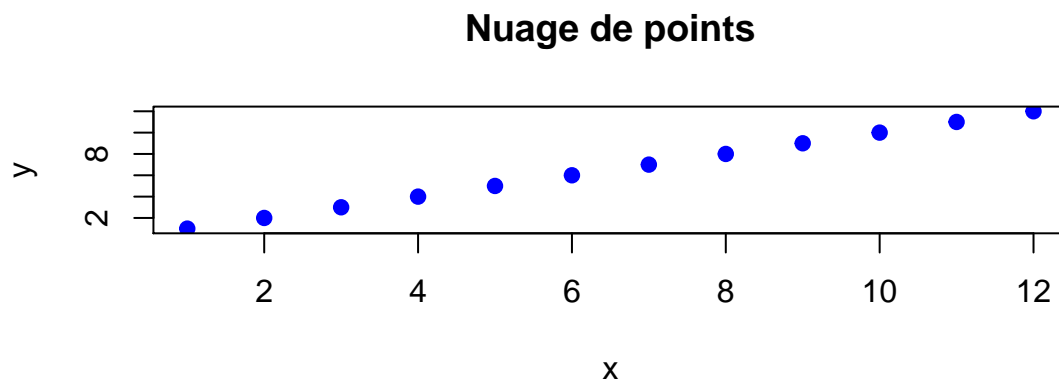
Septième commandement: Tu utiliseras `quantile()` pour trouver les quantiles.

```
y = seq(1:100)
quantile(y,c(.05,.95))
```

```
##      5%    95%
##  5.95 95.05
```

Huitième commandement: `plot()` pourra te servir à tracer un nuage de points.

```
x=seq(1:12)
y=seq(1:12)
plot(x,y,main = "Nuage de points" ,xlab="x", ylab="y", pch=19, col="blue")
```



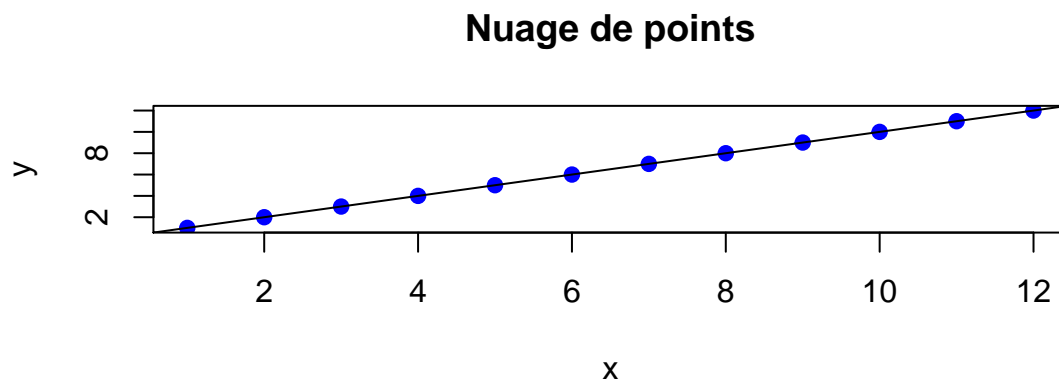
Neuvième commandement: Tu prôneras l'usage de `reg()` pour la régression linéaire.

```
reg1 <- lm(y~x)
summary(reg1)
```

```
## Warning in summary.lm(reg1): essentially perfect fit: summary may be
## unreliable
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.198e-16 -2.914e-16 -1.103e-16  1.118e-16  8.776e-16
##
## Coefficients:
##              Estimate Std. Error  t value Pr(>|t|)
## (Intercept)  0.000e+00  2.602e-16  0.000e+00      1
## x            1.000e+00  3.535e-17  2.829e+16 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.228e-16 on 10 degrees of freedom
## Multiple R-squared:  1, Adjusted R-squared:  1
## F-statistic: 8.001e+32 on 1 and 10 DF, p-value: < 2.2e-16
```

```
plot(x,y,main = "Nuage de points",xlab="x", ylab="y", pch=19, col="blue")
abline(reg1)
```



Dixième commandement: La création d'une fonction fera de toi l'architecte de tes plus grandes édifications.

```
# Fonction sommant les éléments d'un vecteur (la fonction sum() fait la même chose).
# x: Vecteur à sommer.
Gauss <- function(x){
  somme_vecteur = x + rev(x)
  somme = (length(somme_vecteur)*somme_vecteur[1])/2
  return(somme)
}
Gauss(seq(1:100))
```

```
## [1] 5050
```